

# ANALISA KINERJA DATABASE DAN IMPLEMENTASI CACHE REDIS PADA WEB SERVICE LUMEN

<sup>1</sup>M.Budi Hartanto, <sup>2</sup>Teuku Muhammad Fawa'ati, <sup>3</sup>Doni Eko Hendro P,  
<sup>1,2,3</sup>Universitas Mitra Indonesia  
e-mail : <sup>1</sup>budi.hartanto@umitra.ac.id, <sup>2</sup>teuku@umitra.ac.id, <sup>3</sup>donieko@umitra.ac.id

---

## Abstract

Penelitian ini bertujuan untuk melakukan analisis kinerja terhadap database dan implementasi cache Redis pada web service menggunakan framework Lumen. Penelitian ini dilakukan dengan fokus pada peningkatan responsivitas, efisiensi penggunaan sumber daya, dan skalabilitas aplikasi. Metode eksperimental digunakan untuk mengukur kinerja sebelum dan sesudah implementasi cache Redis. Implementasi cache Redis dilakukan dengan mempertimbangkan karakteristik aplikasi dan pola akses data. Pengukuran kinerja melibatkan waktu respons, latensi, dan frekuensi hit dan miss cache. Data dikumpulkan melalui alat-alat pengukuran performa dan analisis statistik.

Hasil analisis menunjukkan bahwa implementasi cache Redis berhasil meningkatkan waktu respons secara signifikan, mengurangi beban database, dan meningkatkan efisiensi penggunaan sumber daya. Frekuensi hit cache yang tinggi menunjukkan efektivitas cache dalam menyimpan dan mengambil data yang sering diminta.

Penelitian ini memberikan wawasan mendalam tentang dampak implementasi cache Redis pada web service Lumen dan menyediakan dasar untuk pemeliharaan, pengembangan, dan peningkatan kinerja berkelanjutan. Rekomendasi termasuk pemeliharaan rutin, pemantauan terus-menerus, dan penyesuaian konfigurasi Redis untuk meningkatkan performa masa depan.

**Keywords:** Analisis Kinerja, Database, Cache Redis, Web Service, Lumen, Responsivitas, Efisiensi Sumber Daya, Skalabilitas.

---

## I. PENDAHULUAN

Dalam ekosistem pengembangan perangkat lunak yang terus berkembang, performa dan responsivitas aplikasi web service menjadi krusial untuk memberikan pengalaman pengguna yang optimal. Analisis kinerja database dan implementasi cache redis pada web service lumen adalah langkah strategis untuk meningkatkan efisiensi, mengurangi waktu respons, dan memastikan keandalan operasional.

Lumen, kerangka kerja mikro yang berbasis pada laravel, menyediakan solusi yang ringan dan efisien untuk pembuatan web service. Dengan pertumbuhan volume pengguna dan permintaan yang semakin tinggi, perluasan fungsionalitas, dan peningkatan kompleksitas aplikasi, performa menjadi fokus utama untuk menjaga kepuasan pengguna.

Latar belakang masalah diantaranya: peningkatan beban database:dengan meningkatnya jumlah pengguna dan transaksi, beban pada database lumen dapat mengalami peningkatan signifikan, mempengaruhi waktu respons dan keandalan sistem. Waktu respons yang lambat:waktu respons yang lambat dapat merugikan pengalaman pengguna dan mengurangi daya saing web service di pasar yang kompetitif. Peningkatan kompleksitas query:pertumbuhan aplikasi dapat menyebabkan kompleksitas query yang tinggi, memerlukan evaluasi untuk memastikan efisiensi eksekusi.Optimalisasi penggunaan sumber daya:pengoptimalan penggunaan sumber daya server dan database menjadi penting untuk mendukung skalabilitas dan kinerja yang optimal.

Tujuan analisis dan implementasi: tujuan analisis dan implementasi: meningkatkan waktu respons: analisis kinerja database dan implementasi redis cache bertujuan untuk mengurangi waktu respons dengan meminimalkan waktu akses data.Optimalisasi query dan indeks:mengidentifikasi dan mengoptimalkan query serta pengindeksan untuk meningkatkan efisiensi eksekusi.pengurangan beban database:melalui implementasi redis cache, tujuannya adalah mengurangi beban pada database dengan menyimpan data yang sering diakses di dalam cache.peningkatan keandalan dan skalabilitas:

Meningkatkan keandalan sistem dan mendukung skalabilitas melalui strategi caching yang efektif.manfaat implementasi cache redis:akses data cepat:redis, sebagai database cache in-memory, memberikan akses data yang cepat dan efisien.pengurangan latensi:mengurangi latensi dengan menyimpan hasil query yang sering dieksekusi di dalam cache, menghindari akses ke database utama.

Skalabilitas yang Ditingkatkan:redis memungkinkan skema clustering, mendukung skalabilitas horizontal untuk menangani pertumbuhan yang cepat.Manajemen session yang efisien:implementasi redis dapat digunakan untuk manajemen session, meningkatkan kecepatan autentikasi dan otorisasi.analisis kinerja database dan implementasi cache redis pada web service lumen merupakan langkah strategis dalam memastikan keberlanjutan dan performa optimal.

Dengan memahami beban bekerja aplikasi, kompleksitas query, dan kebutuhan pengguna, organisasi

dapat mengambil tindakan proaktif untuk meningkatkan efisiensi dan kecepatan layanan. Penelitian dan implementasi ini menjadi dasar yang kokoh untuk memenuhi tuntutan pasar yang terus berkembang dan memastikan keunggulan kompetitif dalam dunia digital yang dinamis.

## II. LANDASAN TEORI

### 2.1 Lumen

Lumen adalah micro-framework PHP yang dikembangkan oleh Taylor Otwell, pencipta Laravel. Lumen dirancang untuk memberikan kinerja tinggi dan kemudahan pengembangan, terutama untuk aplikasi berbasis mikro dan layanan API. Berikut adalah beberapa poin utama tentang Lumen: 1. Micro-framework Ringan: Lumen adalah versi ringan dari Laravel, sebuah framework PHP yang populer.

Ini dibuat dengan tujuan menyediakan dasar yang cepat dan efisien untuk pengembangan aplikasi dengan fokus pada kinerja tinggi. 2. Kinerja Tinggi: Dibandingkan dengan Laravel, Lumen lebih terfokus pada kinerja tinggi. Beberapa fitur yang mungkin tidak selalu diperlukan dalam proyek berbasis mikro dihilangkan untuk memberikan waktu respons yang lebih cepat. 3. Fokus pada API dan Aplikasi Berbasis Mikro: Lumen sangat cocok untuk membangun layanan API dan aplikasi berbasis mikro. Ini menyediakan fitur-fitur yang diperlukan untuk membangun API dengan cepat, seperti routing yang sederhana dan dukungan untuk format respon JSON. 4. Dukungan Eloquent ORM: Lumen mendukung Eloquent ORM, yang memungkinkan pengembang berinteraksi dengan basis data menggunakan model dan query builder.

Ini memberikan antarmuka yang ekspresif untuk mengelola data. 5. Meskipun lebih ringan, Lumen tetap terintegrasi dengan ekosistem Laravel. Ini berarti bahwa kode yang ditulis dalam Lumen dapat dengan mudah dipindahkan atau ditingkatkan ke Laravel jika proyek berkembang dan memerlukan fitur lebih lanjut. 6. Lumen cocok untuk proyek-proyek berukuran kecil hingga menengah yang membutuhkan dasar pengembangan yang kuat tanpa kelebihan fitur dari Laravel.

(<https://lumen.laravel.com/>, <https://github.com/laravel/lumen>, <https://laravel.com/docs>)

### 2.2 Database

Dalam bukunya yang terkenal, "Database System Concepts," Abraham Silberschatz, Henry F. Korth, dan S. Sudarshan memberikan definisi dan konsep dasar mengenai database. Berikut adalah beberapa kutipan dan konsep yang dapat diambil dari buku tersebut: Database adalah kumpulan data yang saling terkait yang disimpan bersama-sama dan diorganisir agar dapat diakses, dikelola, dan diperbarui. Sistem Manajemen Database (DBMS) adalah perangkat lunak yang menyediakan cara untuk menyimpan, mengatur, dan mengelola data dalam database.

Tujuan utama dari database adalah menyediakan metode untuk menyimpan dan mengambil data dengan cepat dan efisien dan memberikan cara untuk menyusun data sehingga dapat diakses dan dimanipulasi dengan mudah. Buku ini juga membahas Distributed Databases, yaitu database yang tersebar di beberapa lokasi fisik dan diakses melalui jaringan. "Transaksi" adalah konsep penting yang dibahas, di mana transaksi dianggap sebagai operasi yang mengakses atau memanipulasi data dalam database.

### 2.3 Cache

Dalam buku "Computer Organization and Design," David A. Patterson dan John L. Hennessy (2022) membahas cache sebagai komponen kunci dalam sistem komputer yang bertujuan meningkatkan kinerja dengan menyediakan penyimpanan yang lebih cepat dan lebih dekat dengan unit pemrosesan (CPU). Berikut adalah beberapa konsep tentang cache dari perspektif buku tersebut: Cache adalah tingkat penyimpanan kecil yang menyimpan sebagian dari data dan instruksi yang sering digunakan. Cache berfungsi sebagai penyangga antara penyimpanan utama (RAM) dan unit pemrosesan (CPU).

Tujuan cache adalah menyediakan akses yang lebih cepat terhadap data dan instruksi yang sering digunakan oleh CPU. Dengan menyimpan salinan data yang sering diakses di cache, waktu yang diperlukan untuk membaca data dari penyimpanan utama dapat dikurangi. Prinsip lokalitas sangat ditekankan. Ada dua jenis lokalitas: lokalitas spasial (data yang berdekatan cenderung diakses bersama) dan lokalitas temporal (data yang baru saja diakses cenderung diakses lagi dalam waktu dekat). Cache dirancang dengan prinsip ini untuk meningkatkan efisiensi. Cache bekerja dengan konsep "hit" dan "miss." Jika data yang diminta oleh CPU ada di cache, itu disebut "hit." Jika tidak, itu disebut "miss," dan data harus diambil dari penyimpanan utama. Cache umumnya diorganisir dalam beberapa tingkat (Level 1, Level 2, dll.), di mana Level 1 (L1) cache lebih kecil dan lebih cepat, sementara Level 2 (L2) cache lebih besar dan sedikit lebih lambat.

### 2.4 Redis

Redis adalah sebuah proyek perangkat lunak open-source yang dikembangkan oleh Salvatore Sanfilippo, yang sering dikenal dengan nama pengguna "antirez". Dia adalah pencipta utama Redis dan kontributor kunci dalam pengembangan proyek tersebut. Redis pertama kali dirilis pada tahun 2009, dan sejak itu telah berkembang menjadi salah satu sistem penyimpanan data berkinerja tinggi yang paling populer di dunia, terutama dikenal karena kecepatannya yang tinggi dan dukungannya untuk berbagai struktur data.

## 2.5 Web Service

Web service adalah suatu layanan atau aplikasi yang disediakan oleh suatu sistem komputer melalui jaringan, biasanya melalui internet. Layanan ini dirancang untuk berinteraksi dengan aplikasi atau sistem lain, memungkinkan pertukaran data dan fungsionalitas antar sistem yang berbeda. Komunikasi Melalui Jaringan: Web service memungkinkan komunikasi antar aplikasi atau sistem melalui jaringan, seperti internet.

Protokol yang umum digunakan untuk komunikasi web service termasuk HTTP (Hypertext Transfer Protocol) dan HTTPS (HTTP Secure). Format Pertukaran Data: Pertukaran data antar sistem dalam web service biasanya menggunakan format standar seperti XML (eXtensible Markup Language) atau JSON (JavaScript Object Notation).

Format ini memungkinkan struktur data yang terstruktur dan dapat dibaca oleh manusia dan mesin. Interoperabilitas: Web service mendukung interoperabilitas, artinya mereka dapat digunakan oleh berbagai jenis platform, bahasa pemrograman, dan sistem operasi. Hal ini dicapai dengan mengadopsi standar tertentu untuk komunikasi dan format data.

Protokol dan Standar: Web service dapat menggunakan berbagai protokol dan standar, seperti SOAP (Simple Object Access Protocol), REST (Representational State Transfer), dan WSDL (Web Services Description Language), untuk mendefinisikan antarmuka dan spesifikasi layanan. Dapat Diakses dan Digunakan oleh Aplikasi Lain: Web service menyediakan antarmuka yang memungkinkan aplikasi lain untuk mengakses fungsionalitas atau data yang disediakan oleh layanan tersebut. Ini memungkinkan integrasi yang efisien antara aplikasi yang berbeda.

## III. METODELOGI PENELITIAN

### 3.1 Jenis Penelitian

- Penelitian Eksperimental:

Deskripsi: Penelitian eksperimental dapat melibatkan pengumpulan data kinerja dengan mengimplementasikan perubahan atau peningkatan spesifik pada sistem, seperti implementasi cache Redis.

Tujuan: Menilai efektivitas perubahan tersebut dalam meningkatkan kinerja sistem secara langsung.

Metode: Pengukuran waktu respons, throughput, latensi, dan metrik kinerja lainnya sebelum dan setelah implementasi cache Redis.

- Penelitian Kuantitatif:

Deskripsi: Mengumpulkan dan menganalisis data kuantitatif terkait dengan kinerja database dan cache Redis, seperti waktu respons, frekuensi hit dan miss cache, dan beban server.

Tujuan: Menyediakan pemahaman yang mendalam tentang dampak implementasi cache pada kinerja aplikasi.

Metode: Pengukuran kuantitatif dengan menggunakan alat pengukuran performa dan monitoring.

- Penelitian Kasus:

Deskripsi: Melibatkan studi kasus terhadap implementasi cache Redis pada web service Lumen tertentu.

Tujuan: Memberikan wawasan mendalam tentang pengalaman dan hasil spesifik yang ditemukan selama implementasi cache.

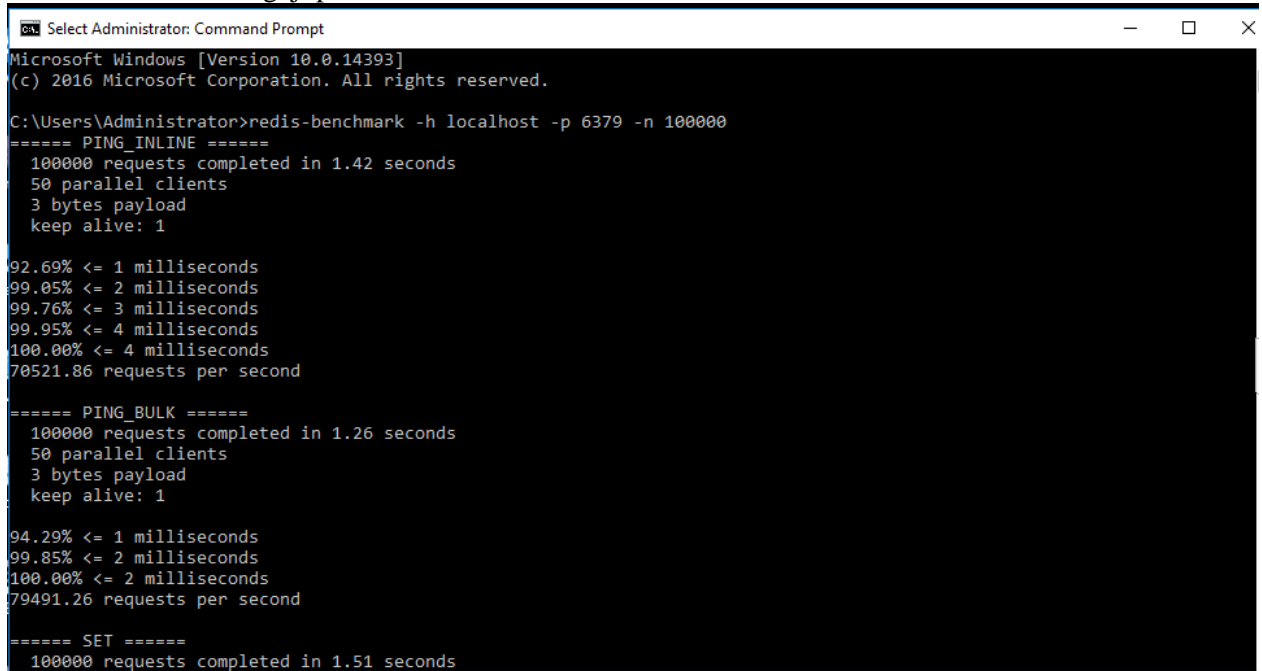
Metode: Pengumpulan data melalui observasi, wawancara, dan analisis dokumentasi.

### 3.2 Metode Penelitian

Pengujian kinerja Redis cache melibatkan beberapa langkah untuk mengevaluasi seberapa baik cache Redis berfungsi dalam mengoptimalkan akses data dan meningkatkan performa aplikasi. Berikut adalah beberapa metode pengujian cache Redis yang umum digunakan:

Menggunakan redis-benchmark:

- a) Seperti yang telah dibahas sebelumnya, Anda dapat menggunakan redis-benchmark yang disediakan oleh Redis untuk menguji performa server Redis.



```
Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>redis-benchmark -h localhost -p 6379 -n 100000
===== PING_INLINE =====
100000 requests completed in 1.42 seconds
50 parallel clients
3 bytes payload
keep alive: 1

92.69% <= 1 milliseconds
99.05% <= 2 milliseconds
99.76% <= 3 milliseconds
99.95% <= 4 milliseconds
100.00% <= 4 milliseconds
70521.86 requests per second

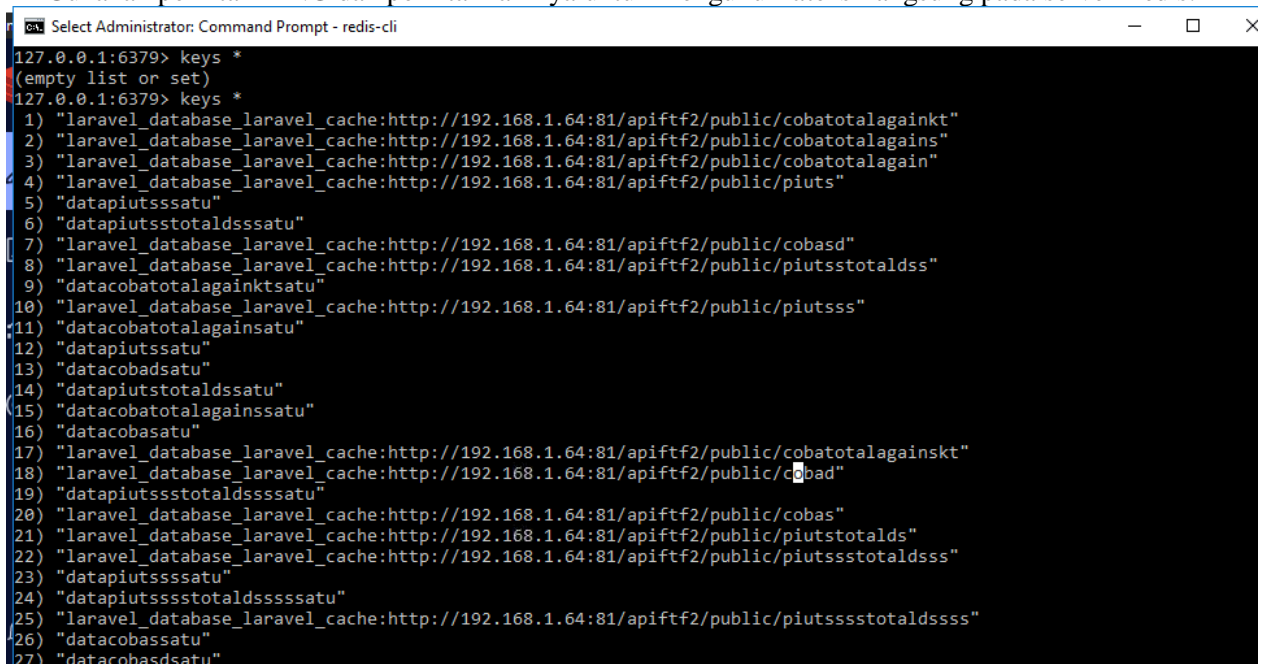
===== PING_BULK =====
100000 requests completed in 1.26 seconds
50 parallel clients
3 bytes payload
keep alive: 1

94.29% <= 1 milliseconds
99.85% <= 2 milliseconds
100.00% <= 2 milliseconds
79491.26 requests per second

===== SET =====
100000 requests completed in 1.51 seconds
```

- b) Pengukuran Latensi dengan Perintah Redis:

Gunakan perintah PING dan perintah lainnya untuk mengukur latensi langsung pada server Redis.



```
Select Administrator: Command Prompt - redis-cli
127.0.0.1:6379> keys *
(empty list or set)
127.0.0.1:6379> keys *
1) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobatotalagaintk"
2) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobatotalagains"
3) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobatotalagains"
4) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutss"
5) "datapiutssssatu"
6) "datapiutssstotaldssssatu"
7) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobas"
8) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutssstotaldss"
9) "datacobatotalagaintksatu"
10) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutsss"
11) "datacobatotalagainsatu"
12) "datapiutssatu"
13) "datacobadsatu"
14) "datapiutssstotaldssssatu"
15) "datacobatotalagainsatu"
16) "datacobasatu"
17) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobatotalagainskt"
18) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobad"
19) "datapiutssstotaldssssatu"
20) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/cobas"
21) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutssstotaldss"
22) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutssstotaldss"
23) "datapiutssssatu"
24) "datapiutssstotaldssssatu"
25) "laravel_database_laravel_cache:http://192.168.1.64:81/apiftf2/public/piutssstotaldss"
26) "datacobassatu"
27) "datacobadsatu"
```

- c) Pengujian dengan menggunakan fungsi GET HTTP Responses dengan redis

New Collection / use redis

GET http://103.140.188.145:81/apiftf2/public/coba Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 869 ms 28.07 KB Save Response

Pretty Raw Preview Visualize JSON

```
9      "KECAMATAN": "ANAK RATU AJI",  
10     "KELURAHAN": "GEDUNG RATU",  
11     "JML_SPPT": "1045",  
12     "JUMLAH": 45729401
```

d) Pengujian dengan menggunakan fungsi GET HTTP Responses dengan non-redis

New Collection / no redis

GET http://103.140.188.145:81/apiftf/public/coba Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 48.78 s 28.07 KB Save Response

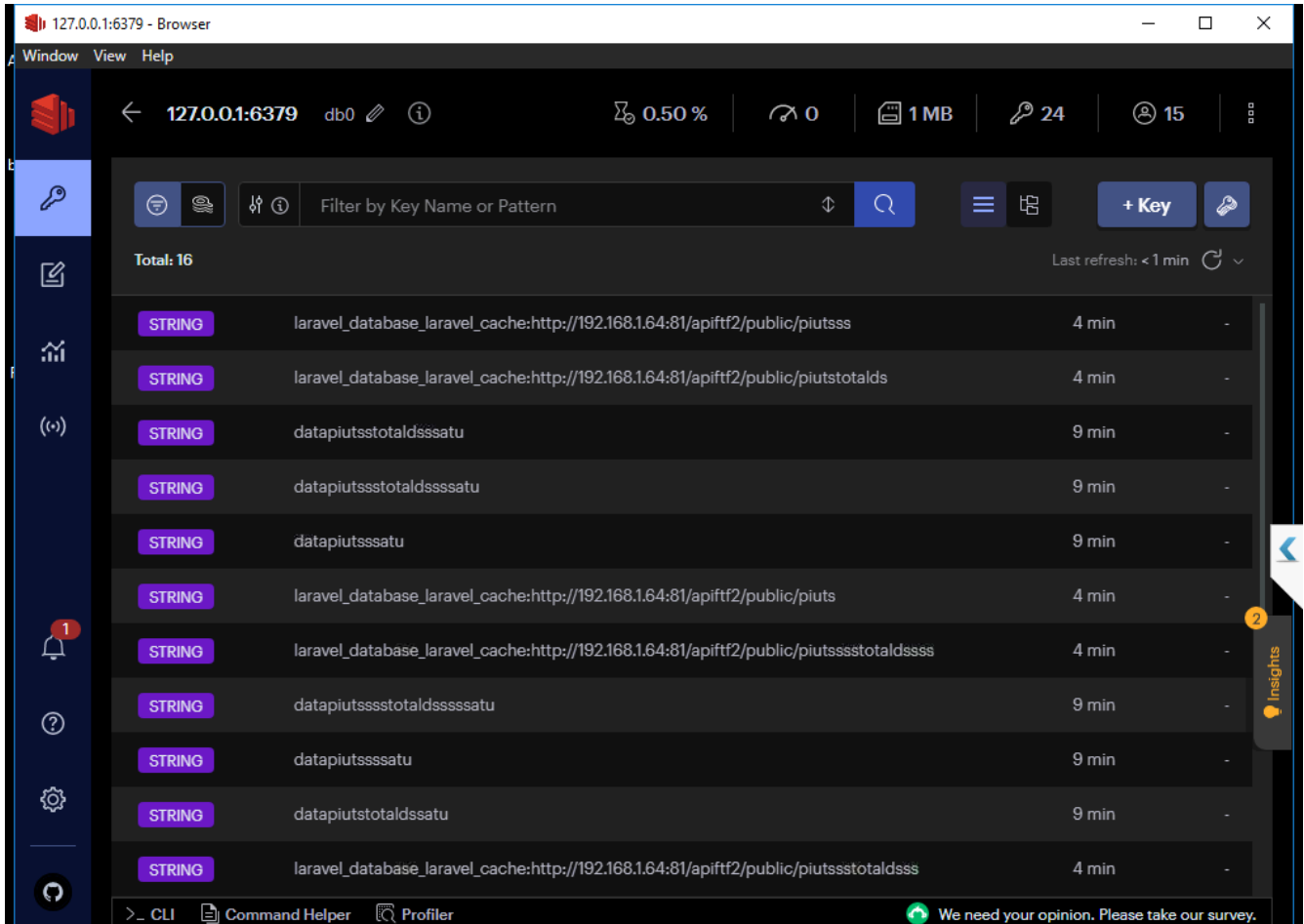
Pretty Raw Preview Visualize JSON

```
1  
2  {  
3    "KECAMATAN": "ANAK RATU AJI",  
4    "KELURAHAN": "BANDAR PUTIH TUA",  
5    "JML SPPT": "1045"
```

Sebuah analisis kinerja database dan implementasi cache Redis pada web service Lumen (yang merupakan framework PHP untuk membangun layanan web) dapat memberikan wawasan yang berharga terkait dengan efisiensi, responsifitas, dan skalabilitas aplikasi. Berikut adalah contoh hasil dan pembahasan yang mungkin muncul dari analisis tersebut.

### 3.3 Performa Database

Gambar 1. Monitoring performa database dalam menggunakan redis



Gambar 2. Memantau latensi saat berinteraksi dengan cache Redis untuk memastikan bahwa respons cepat dan konsisten

```
mingw64:/f/wamp64/www/pajak-realisasi2
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)...

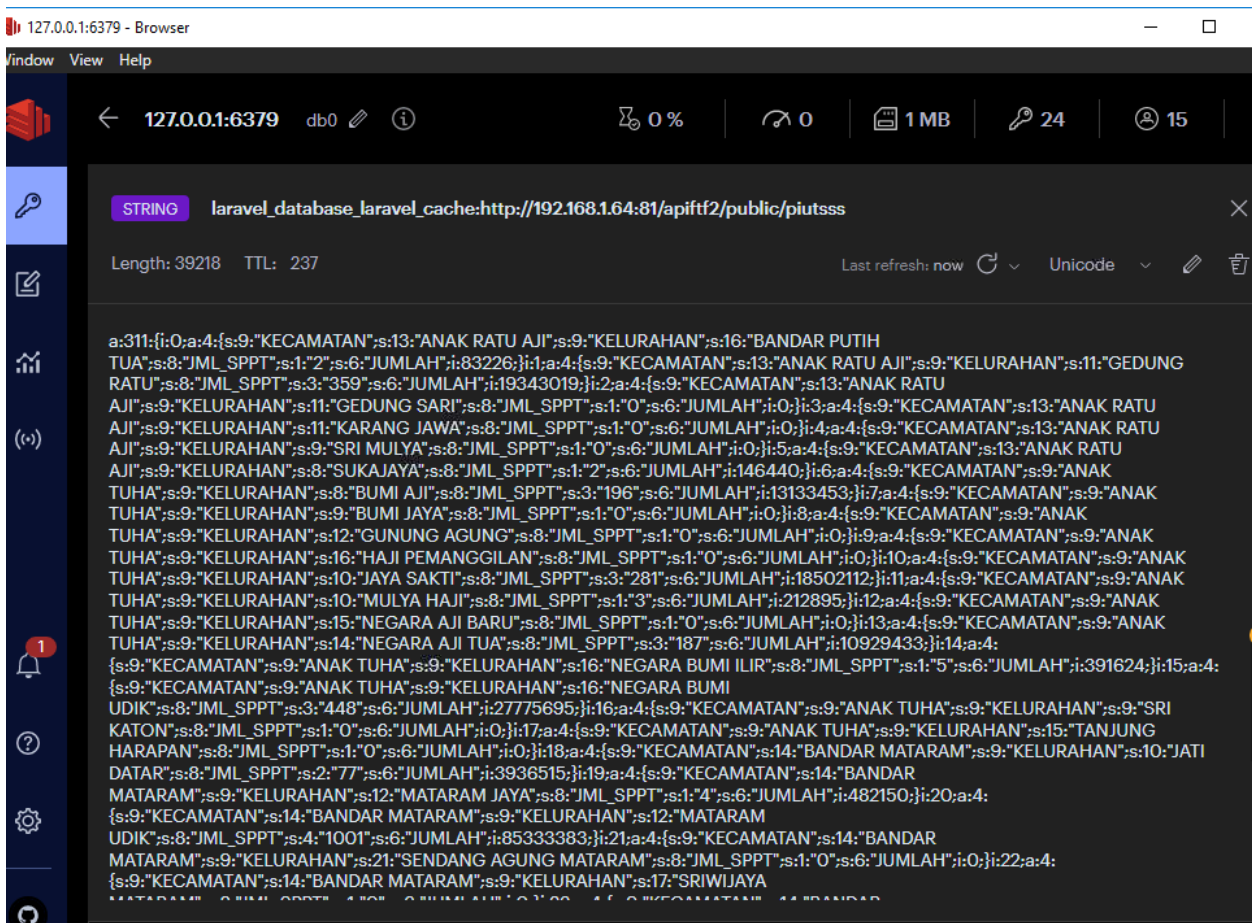
Server Software:
Server Hostname:      localhost
Server Port:          8000

Document Path:        /your-route
Document Length:      6609 bytes

Concurrency Level:    10
Time taken for tests:  2.514 seconds
Complete requests:    9
Failed requests:      0
Non-2xx responses:    9
Total transferred:    61650 bytes
HTML transferred:    59481 bytes
Requests per second:  3.58 [#/sec] (mean)
Time per request:     2793.416 [ms] (mean)
Time per request:     279.342 [ms] (mean, across all concurrent requests)
Transfer rate:        23.95 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:  0    0   0.4      0     1
Processing: 153  664 413.1    718  1292
Waiting:  151  662 413.6    717  1291
```

Gambar 3. Hasil dari loading data berupa data json





Gambar 4. Statistik data 'info' redis

```
127.0.0.1:6379> info
# Server
redis_version:3.0.504
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:a4f7a6e86f2d60b3
redis_mode:standalone
os:Windows
arch_bits:64
multiplexing_api:WinSock_IOCP
process_id:832
run_id:87e5ca2fb6ebe801557ca03ec0b79700b846da6c
tcp_port:6379
uptime_in_seconds:358921
uptime_in_days:4
hz:10
lru_clock:10670042
config_file:

# Clients
connected_clients:1
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0

# Memory
used_memory:26292696
used_memory_human:25.07M
used_memory_rss:26255056
used_memory_peak:30360008
```

Gambar 5. Implementasi Redis dalam lumen

```
// Store the data in Redis cache for future use with a specific
expiration time (e.g., 10 minutes)
    Redis::setex($cacheKey, 600, json_encode($reals));
} else {
    // Data is available in Redis cache, decode it
    $reals = json_decode($reals);
}

return response()->json($reals);
```



## V. KESIMPULAN

Berdasarkan hasil analisa yang telah dilakukan maka dapat ditarik kesimpulan kinerja database dan implementasi cache Redis pada web service Lumen akan tergantung pada hasil spesifik yang ditemukan selama proses analisis. Namun, berikut adalah beberapa kesimpulan umum yang mungkin dapat diambil:

### 1. Efektivitas Implementasi Cache Redis:

Jika tingkat hit cache tinggi dan tingkat miss rendah, ini menunjukkan bahwa implementasi cache Redis berhasil dalam menyimpan dan mengambil data dengan efisien, mengurangi beban pada database.

### 2. Peningkatan Waktu Respons:

Jika ditemukan peningkatan signifikan dalam waktu respons untuk permintaan tertentu setelah implementasi cache, ini menunjukkan bahwa cache Redis efektif dalam meningkatkan kinerja aplikasi dengan menyimpan hasil yang sering diminta.

### 3. Reduksi Beban Database:

Jika beban database berkurang secara signifikan, terutama untuk kueri yang intensif, ini menunjukkan bahwa cache Redis telah berhasil mengurangi jumlah permintaan yang harus diarahkan ke database, meningkatkan efisiensi penggunaan sumber daya.

### 4. Optimalisasi Penggunaan Sumber Daya:

Jika penggunaan sumber daya seperti CPU dan memori database berkurang, ini menunjukkan bahwa implementasi cache Redis membantu mengoptimalkan penggunaan sumber daya dan dapat mendukung skalabilitas yang lebih baik.

### 5. Identifikasi dan Solusi Masalah:

Jika ada masalah performa tertentu yang diidentifikasi selama analisis, langkah-langkah perbaikan dan pemeliharaan yang disarankan dapat membantu dalam meningkatkan kinerja keseluruhan.

### 6. Rekomendasi untuk Masa Depan:

Berikan rekomendasi untuk pemeliharaan rutin, pemantauan terus-menerus, dan kemungkinan peningkatan lebih lanjut dalam implementasi cache Redis atau konfigurasi database.

### 7. Ketahanan dan Skalabilitas:

Jika analisis menunjukkan bahwa sistem mampu menangani lonjakan beban dan masih memberikan waktu respons yang baik, ini adalah indikasi bahwa web service Lumen bersifat tangguh dan dapat diskalakan.

### 8. Pemahaman Holistik:

Kesimpulan seharusnya memberikan gambaran holistik tentang kinerja keseluruhan sistem, menggabungkan efektivitas cache Redis, performa database, dan respons aplikasi secara keseluruhan.

Kesimpulannya cache Redis telah memengaruhi kinerja web service Lumen dan memberikan dasar untuk pengambilan keputusan lebih lanjut dalam pemeliharaan dan pengembangan aplikasi. Selain itu, pemantauan dan evaluasi kinerja secara teratur sangat dianjurkan untuk memastikan kesinambungan dan perbaikan berkelanjutan.

## REFERENCES

- L. D. Putri, "Evaluasi User Interface Web Commerce Menggunakan Aturan Eight Golden Rules," *Indones. J. Appl. Informatics*, vol. 5, no. 2, p. 94, 2022, doi: 10.20961/ijai.v5i2.41935.
- N. A. O. Saputri and A. Alvin, "Measurement of User Satisfaction Level in the Bina Darma Information Systems Study Program Portal Using End User Computing Satisfaction Method," *J. Inf. Syst. Informatics*, vol. 2, no. 1, pp. 154–162, 2020, doi: 10.33557/journalisi.v2i1.43.
- N. M. Sulistyawati, "Kepuasan Pelanggan Restoran Indus," *E- J. Manaj. Unud*, vol. 4, no. 8, pp. 2318–2332, 2015, [Online]. Available: <http://journal.uta45jakarta.ac.id/index.php/jbsuta/article/viewFile/959/667>
- Y. S. Jamilah and A. C. Padmasari, "Perancangan User Interface Dan User Experience Aplikasi Say.Co," *J. Desain Komun. Vis.*, vol. 9, no. 2, pp. 73–78, 2022, [Online]. Available: <https://ojs.unm.ac.id/tanra/article/view/29458>
- Rahartri, "'Whatsapp' Media Komunikasi Efektif Masa Kini (Studi Kasus Pada Layanan Jasa Informasi Ilmiah di Kawasan Puspipetek)," *Visi Pustaka*, vol. 21, no. 2, pp. 147–156, 2019.
- L. S. Ningsih, F. Fitriyani, Z. H. Hasibuan, and ..., "Pemanfaatan Media Whatshapp Sebagai Sarana Komunikasi Bagi Pustakawan," *Maktab. J. ...*, vol. 2275, pp. 1–12, 2022, [Online]. Available: <https://ummaspul.e-journal.id/RMH/article/view/4436>
- A. Nasution, M. S. I. Lubis, M. Luthfi, and ..., "Komunikasi Melalui Aplikasi Whatsapp Dalam Rangka Pembelajaran Anak Sekolah Dasar Masa Pandemi Covid-19 Di Lingkungan Medan ...," ... *J. Ilm. ...*, vol. 18, 2021, [Online]. Available: <https://komunikologi.esaunggul.ac.id/index.php/KM/article/view/356>

- C. O. S. Patricia, "No主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title," vol. 3, no. 2, p. 6, 2021.
- M. R. Fadli, "Memahami desain metode penelitian kualitatif," *Humanika*, vol. 21, no. 1, pp. 33–54, 2021, doi: 10.21831/hum.v21i1.38075.
- M. W. Richard Romario Samuel Rawis, S. L. H. V. Joyce. Lopian and U. S. Ratulangi, "Service Quality, Brand," *Richard Romario Samuel Rawis, S. L. H. V. Joyce. Lopian, Magdal. Wullur Univ. Sam Ratulangi*, vol. 7, no. 2, pp. 464–475, 2020.
- F. Susanto, *Pengenalan Sistem Pendukung Keputusan*. Deepublish, 2020.
- S. Asmawati et al., *Sistem Pendukung Keputusan*. Media Sains Indonesia, 2022.
- Kieso, *Akuntansi Intermediate, Edisi Kedua Belas*. Jakarta: Erlangga, 2019.
- Handoko, *Manajemen Production Planning And Inventory Control*. Jakarta: PT. Gramedia Pustaka Umum, 2018.
- M. Hasanah, N. H. Harani, and N. Riza, *Implementasi Barcode Dan Algoritma Regresi Linear Untuk Memprediksi Data Persediaan Barang*. Kreatif, 2020.
- Ghozali, *Aplikasi Analisis Multivariate Dengan Program IBM SPSS 23.(edisi kedelapan)*. Semarang: Badan Penerbit Universitas Diponegoro, 2016.
- D. Krisbiantoro, M. Kom, P. D. Abda'u, and M. Kom, *Dasar Pemrograman Web Dengan Bahasa HTML, PHP, dan Database MySQL*, vol. 1. Zahira Media Publisher, 2021.
- A. Rozaq, *Konsep Perancangan Sistem Informasi Bisnis Digital*. Poliban Press, 2020.
- P. Duan, Y. Cao, Y. Wang, and P. Yin, "Bibliometric Analysis of Coastal and Marine Tourism Research from 1990 to 2020," *Journal of Coastal Research*, vol. 38, no. 1. Global Eksekutif Teknologi, pp. 229–240, 2022. doi: 10.2112/JCOASTRES-D-20-00171.1.
- A. M. A. Rusdy, P. Purnawansyah, and H. Herman, "Penerapan Metode Regresi Linear Pada Prediksi Penawaran dan Permintaan Obat Studi Kasus Aplikasi Point Of Sales," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 3, no. 2, pp. 121–126, 2022.
- R. Destriana et al., *Diagram UML Dalam Membuat Aplikasi Android Firebase" Studi Kasus Aplikasi Bank Sampah"*. Deepublish, 2021.
- H. S. Wibowo, *Belajar HTML untuk Pemula: Panduan Lengkap untuk Membuat Halaman Web yang Menarik*. Pesona Bahasa, 2023.
- N. Ahmad et al., *Analisa & Perancangan Sistem Informasi Berorientasi Objek*. Penerbit Widina, 2022.
- E. S. Pane, I. Ishak, and A. H. Nasyuha, "Sistem Pendukung keputusan Menentukan Persediaan Obat Pada Rumah Sakit Mitra Sejati Dengan Menggunakan Metode Fuzzy Tsukamoto," *J. Cyber Tech*, vol. 2, no. 6, 2019.